MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2018/2019

ECP2216 – MICROCONTROLLER AND MICROPROCESSOR SYSTEMS

(TE / RE)

13 OCTOBER 2018
2.30 p.m – 4.30 p.m
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 10 pages with 5 Questions only.

2. Attempt **ALL FIVE COMPULSORY** questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

Question 1

(a) State the FOUR major components in a computer system. [2 marks]

(b) Given a memory block has 34 address lines and 64 data lines. Determine the memory capacity of this memory block. Express your answer in the unit of bytes. *(You are required to show out all working steps)*

[3 marks]

(c) Convert a 9-bit binary number, $Y = 1\ 1010\ 1100_2$ into decimal number for the following scenarios. *(You are required to show out all working steps)*

(i) ***Scenario1:*** Y represents an unsigned number. [2 marks]

(ii) ***Scenario2:*** Y represents a signed number. [3 marks]

(d) (i) Intel announced the 80386DX, which is the first 32-bit microprocessor and is a logical extension of the Intel 80286 in year 1985. The 80386DX has six functional units that operate in parallel for pipelined processing. Name these six functional units and highlight which unit provides memory management in protected mode.

[6 marks]

(ii) State the advantage(s) of the dual-core processor technology as compared to two coupled single-core processor system.

[4 marks]

Continued .....

Question 2

(a) Determine the bit addresses which are set to 1 after the execution of the following instruction.

MOV 23H, # 0A5H

[2 marks]

(b) Describe the functions of ***ALE*** and ***PSEN*** pins in 8051.

[2 marks]

(c) Evaluate the contents of Program Status Word (PSW) and Accumulator (A) after the execution of the following assembly language instructions in sequence. Express the contents in **binary** format. (*Assume initial value: A=00H and PSW=00H*)

(i) MOV R7, #88 [1 mark]

(ii) MOV A, #0A9H [1 mark]

(iii) ADD A, R7 [1 mark]

(d) Design an 8051 microcontroller based system which can address contiguous 4Kbytes of external memory space. 1Kbytes of external program memory should occupy the first portion of the memory space followed by 3Kbytes of external data memory. Available memory ICs are 1Kbytes ROM and 1Kbytes RAM.

(i) Identify the number of ROM ICs and RAM ICs required. [1 mark]

(ii) Calculate the size of address bus required. [2 marks]

(iii) Draw and label the system configuration showing the 8051 signal lines to be used for data, address and control buses.

[10 marks]

Continued .....

Question 3

(a) Write a MCS-51 assembly language ***program*** to perform the following tasks:

Step 1: Initialize P2.0 to be a digital input.

Step 2: Delay 1ms.

Step 3: If P2.0 logic level is LOW, call subroutine "CLEARSET",
else call subroutine "SETCLEAR".
In subroutine "CLEARSET", clear P1.0 and set P1.1.
In subroutine "SETCLEAR", set P1.0 and clear P1.1.

Step 4: Repeat Step 2

Assume 12 MHz crystal clock is used.

[15 marks]

(b) Determine the corresponding MCS-51 assembly language instructions for the sequence of machines codes as shown in ***Table Q3 (b).***

Address	Machine Codes
0006H	75H 3BH BEH
0009H	A9H 3BH
000BH	D9H FEH
000DH	22H

Table Q3 (b)

[5 marks]

Continued .....

Question 4

(a) State **ALL** interrupts sources which are available in 8051 microcontroller and arrange them in the order corresponding to their default priority.

[5 marks]

(b) Write a MCS-51 assembly language program with **interrupts** to perform the following task.

An 8051 microcontroller system is designed to use external interrupt INT1 pin which is connected to switch A. This switch is normally high. Whenever switch A goes low, it generates a pulse width of 0.85ms from Port 1 pin 2 using timer 1 interrupt.

[10 marks]

(c) Consider an 8051 microcontroller operates at 11.0592MHz crystal frequency and it needs to be initialized as UART with *SMOD = 1, 8-bit data, 1 stop bit, no parity and 4800 baud rate generated by Timer 1.*

(i) Compute the required reload value for Timer 1 register.

[2 marks]

(ii) Write the MCS-51 assembly language instruction sequence to initialize SCON register, TMOD register and TH1 register.

[3 marks]

Continued .....

Question 5

(a) Table 5(a) shows the bit patterns for each character to decode the seven-segment display.

Table 5(a)

Digit	*h*	*g*	*f*	*e*	*d*	*c*	*b*	*a*
P	0	1	1	1	0	0	1	1
I	0	0	0	0	0	1	1	0
E	0	1	1	1	1	0	0	1

Figure Q5(a) shows an 8051 microcontroller interfaces to two press buttons BTN1 and BTN2 on P1.1 and P1.2 respectively, and a seven-segment LED display device on Port 3.

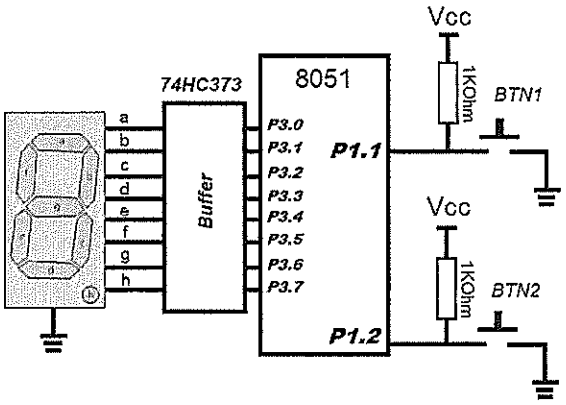


Figure 5(a)

Write a MCS-51 assembly language subroutine called ***PIESubroutine*** to perform the following tasks:

- ***When the subroutine starts, wait for the button press on BTN1.***
- ***Once the BTN1 is pressed, seven-segment LED display will repeatedly display the characters in sequence starting from P, I, and E.*** *(Time duration for each character to be displayed is 0.5 second.)*
- ***The display will be turned off only if the button press on BTN2 occurs.***

NOTE: Assume 0.5 second delay subroutine ***HALFSEC*** *is available.*

[8 marks]

Continued .....

(b) An 8051 microcontroller based ***Automated Hot Latte Maker*** is shown in Figure 5(b) which performs the following process:

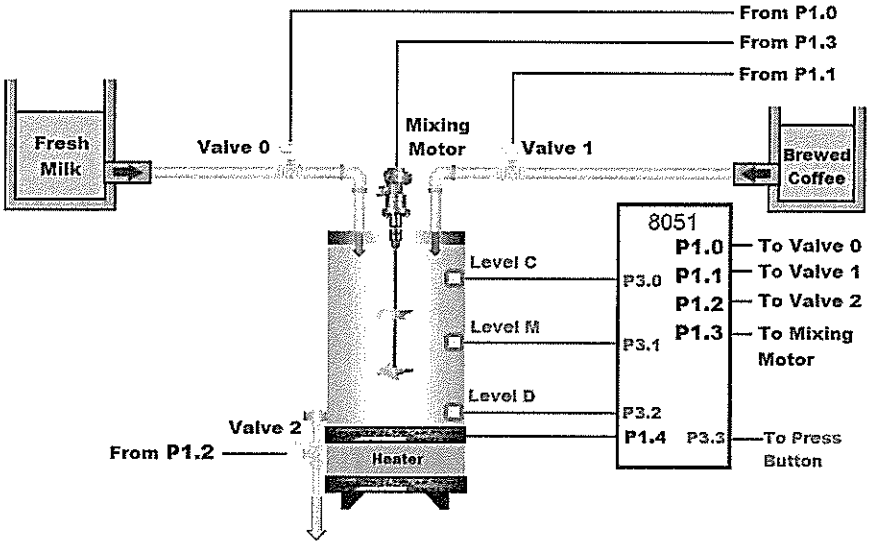


Figure 5(b)

1. ***Initially, all valves, heater and motor are closed.***
2. ***When press button is pressed and released, the container is first filled with fresh milk through a solenoid Valve 0.***
3. ***When the fresh milk reaches Level M, Valve 0 is closed and heater starts the heating process which boils the fresh milk for 3 minutes.***
4. ***Next, the container is filled with brewed coffee via Valve 1 until mixture in the container reaches Level C, Valve 1 is closed.***
5. ***The mixing motor starts the stirring process that last for approximately 1 minutes.***
6. ***Finally, the dispenser Valve 2 opens to dispense the hot latte until Level D is reached. The whole process is repeated again from step 2 onwards.***

The output lines ***P1.0***, ***P1.1*** and ***P1.2*** from 8051 microcontroller provide signals to the solenoid valves. A logical **HIGH** from the lines will open the corresponding valve. The output lines ***P1.3*** and ***P1.4*** from 8051 microcontroller provide signals to the mixing motor and heater respectively which are both activated by a logical **HIGH**. The system has three level sensors and one press button that send signals to input lines ***P3.0*** to ***P3.3***. A logical **HIGH** from the sensor indicates that it has detected the content in the container. A logical **LOW** from the press button indicates that it is pressed. Write a MCS-51 assembly language program to carry out the process. Assume 12MHz crystal frequency is used.

[12 marks]

End of Page

APPENDIX

Special Function Register Formats

Interrupt Enable (IE)

Bit Addr.	AFH	-	-	ACH	ABH	AAH	A9H	A8H
Name	EA	-	-	ES	ET1	EX1	ET0	EX0

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IE.7	EA	Global enable/disable. EA = 1, each individual source is enabled/disabled by setting/clearing its enable bit. EA = 0, disable all interrupts.
IE.6	-	Undefined
IE.5	-	Not implemented in 8051. ET2 for 8052.
IE.4	ES	Serial port interrupt enable bit.
IE.3	ET1	Timer 1 interrupt enable bit.
IE.2	EX1	External interrupt enable bit.
IE.1	ET0	Timer 0 interrupt enable bit.
IE.0	EX0	External interrupt enable bit.

Interrupt Priority (IP)

Bit Addr.	-	-	-	BCH	BBH	BAH	B9H	B8H
Name	-	-	-	PS	PT1	PX1	PT0	PX0

BIT	SYMBOL	FUNCTION (Enable=1, Disable=0)
IP.7	-	Undefined
IP.6	-	Undefined
IP.5	-	Not implemented in 8051. PT2 for 8052.
IP.4	PS	Serial port interrupt priority bit.
IP.3	PT1	Timer 1 interrupt priority bit.
IP.2	PX1	External interrupt priority bit.
IP.1	PT0	Timer 0 interrupt priority bit.
IP.0	PX0	External interrupt priority bit.

Interrupt Vectors

Interrupt Source	Flag	Vector Address
System Reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI & TI	0023H
Timer 2 (8052)	TF2 or EXF2	002BH

Program Status Word (PSW)

Bit Addr.	D7H	D6H	D5H	D4H	D3H	D2H	-	D0H
Name	CY	AC	F0	RS1	RS0	0V	-	P

Serial Control (SCON)

Bit Addr.	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
Name	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

BIT	SYMBOL	FUNCTION
SCON.7	SM0	Serial port mode bit 0 (see Table A.1).
SCON.6	SM1	Serial port mode bit 1 (see Table A.1).
SCON.5	SM2	Serial port mode bit 2; enables multiprocessor communications in modes 2 and 3; RI will not be activated if received 9th bit is 0. In mode 1, if SM2 = 1, then RI will be activated only if a valid stop bit was received. In mode 0, SM2 should be 0.
SCON.4	REN	Receiver enable; must be set to receive characters.
SCON.3	TB8	Transmit bit 8; 9th bit transmitted in modes 2 and 3; set/cleared by software.
SCON.2	RB8	Receive bit 8; 9th bit received.
SCON.1	TI	Transmit interrupt flag; set at end of character transmission; cleared by software.
SCON.0	RI	Receive interrupt flag; set at end of character reception; cleared by software.

Table A.1 The 8051 Serial Port Mode Selection

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift register	Fixed
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fixed
1	1	3	9-bit UART	Variable

Timer Control (TCON)

Bit Addr.	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer-1 overflow flag. Set by hardware on overflow. Cleared by hardware when processor vectors to interrupt routine. Must be cleared by software when not involve interrupt
TCON.6	TR1	Timer-1 run control bit. Set/cleared by software to turn timer/counter on/off.
TCON.5	TF0	Timer-0 overflow flag. Do the same function as TF1 but for Timer-0.
TCON.4	TR0	Timer-0 run control bit. Do the same function as TR1 but for Timer-0.
TCON.3	IE1	External interrupt-1 edge flag. Set by hardware when interrupt-1 falling edge is detected. Cleared by hardware when interrupt is processed.
TCON.2	IT1	Interrupt-1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	External interrupt-1 edge flag. Do the same function as IE1 but for external interrupt-0.
TCON.0	IT0	Interrupt-0 Type control bit. Do the same function as IT1 but for external interrupt-0.

Timer Mode (TMOD)

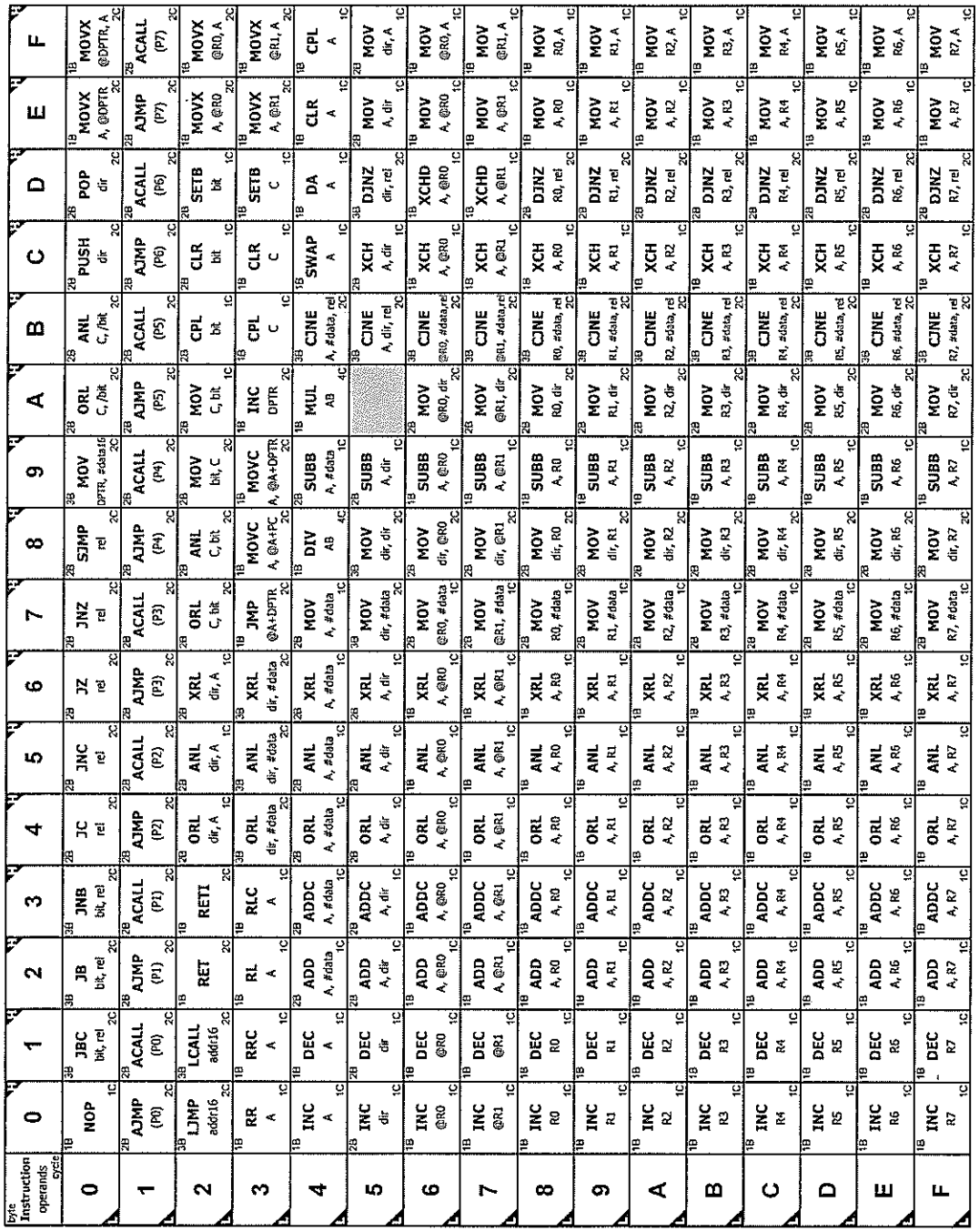
Bit	7	6	5	4	3	2	1	0
Name	GATE	C/T	M1	M0	GATE	C/T	M1	M0

BIT	SYMBOL	FUNCTION
TMOD.7	GATE1	When this bit is set the timer will only run when INT1 (P3.3) is high (hardware control). When this bit is cleared the timer will run regardless of the state of INT1 (software control).
TMOD.6	C/T1	Timer / Counter select bit. C / $\overline{T}$ = 0 → Timer operation. C / $\overline{T}$ = 1 → Counter operation.
TMOD.5	M1	Mode selection bits (see Table A.2). [for timer 1]
TMOD.4	M0	Mode selection bits (see Table A.2). [for timer 1]
TMOD.3	GATE0	Exactly the same function as GATE1 but for Timer0
TMOD.2	C/T0	Exactly the same function as C/T1 but for Timer0
TMOD.1	M1	Mode selection bits (see Table A.2). [for timer 0]
TMOD.0	M0	Mode selection bits (see Table A.2). [for timer 0]

Table A.2 Timer Mode Selection

M1	M0	Timer Mode	Description of Mode
0	0	0	13-bit Timer
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

MCS-51 Opcode Map


byte
Instruction
operands
cycle
0 NOP 1B 1C
1 AJMP (P0) 2B 2C
2 LJMP addr16 3B 3C
3 RR A 1B 1C
4 INC A 1B 1C
5 INC dir 2B 1C
6 INC @R0 1B 1C
7 INC @R1 1B 1C
8 INC R0 1B 1C
9 INC R1 1B 1C
A INC R2 1B 1C
B INC R3 1B 1C
C INC R4 1B 1C
D INC R5 1B 1C
E INC R6 1B 1C
F INC R7 1B 1C
0 JBC bit, rel 3B 2C
1 ACALL (P0) 2B 2C
2 LCALL addr16 3B 3C
3 RRC A 1B 1C
4 DEC A 1B 1C
5 DEC dir 2B 1C
6 DEC @R0 1B 1C
7 DEC @R1 1B 1C
8 DEC R0 1B 1C
9 DEC R1 1B 1C
A DEC R2 1B 1C
B DEC R3 1B 1C
C DEC R4 1B 1C
D DEC R5 1B 1C
E DEC R6 1B 1C
F DEC R7 1B 1C
0 JB bit, rel 3B 2C
1 AJMP (P1) 2B 2C
2 RET 1B 2C
3 RL A 1B 1C
4 ADD A, #data 2B 1C
5 ADD A, dir 2B 1C
6 ADD A, @R0 1B 1C
7 ADD A, @R1 1B 1C
8 ADD A, R0 1B 1C
9 ADD A, R1 1B 1C
A ADD A, R2 1B 1C
B ADD A, R3 1B 1C
C ADD A, R4 1B 1C
D ADD A, R5 1B 1C
E ADD A, R6 1B 1C
F ADD A, R7 1B 1C
0 JNB bit, rel 3B 2C
1 ACALL (P1) 2B 2C
2 RETI 1B 2C
3 RLC A 1B 1C
4 ADDC A, #data 2B 1C
5 ADDC A, dir 2B 1C
6 ADDC A, @R0 1B 1C
7 ADDC A, @R1 1B 1C
8 ADDC A, R0 1B 1C
9 ADDC A, R1 1B 1C
A ADDC A, R2 1B 1C
B ADDC A, R3 1B 1C
C ADDC A, R4 1B 1C
D ADDC A, R5 1B 1C
E ADDC A, R6 1B 1C
F ADDC A, R7 1B 1C
0 JC rel 2B 2C
1 AJMP (P2) 2B 2C
2 ORL dir, A 2B 1C
3 ORL dir, #data 3B 2C
4 ORL A, #data 2B 1C
5 ORL A, dir 2B 1C
6 ORL A, @R0 1B 1C
7 ORL A, @R1 1B 1C
8 ORL A, R0 1B 1C
9 ORL A, R1 1B 1C
A ORL A, R2 1B 1C
B ORL A, R3 1B 1C
C ORL A, R4 1B 1C
D ORL A, R5 1B 1C
E ORL A, R6 1B 1C
F ORL A, R7 1B 1C
0 JNC rel 2B 2C
1 ACALL (P2) 2B 2C
2 ANL dir, A 2B 1C
3 ANL dir, #data 3B 2C
4 ANL A, #data 2B 1C
5 ANL A, dir 2B 1C
6 ANL A, @R0 1B 1C
7 ANL A, @R1 1B 1C
8 ANL A, R0 1B 1C
9 ANL A, R1 1B 1C
A ANL A, R2 1B 1C
B ANL A, R3 1B 1C
C ANL A, R4 1B 1C
D ANL A, R5 1B 1C
E ANL A, R6 1B 1C
F ANL A, R7 1B 1C
0 JZ rel 2B 2C
1 AJMP (P3) 2B 2C
2 XRL dir, A 2B 1C
3 XRL dir, #data 3B 2C
4 XRL A, #data 2B 1C
5 XRL A, dir 2B 1C
6 XRL A, @R0 1B 1C
7 XRL A, @R1 1B 1C
8 XRL A, R0 1B 1C
9 XRL A, R1 1B 1C
A XRL A, R2 1B 1C
B XRL A, R3 1B 1C
C XRL A, R4 1B 1C
D XRL A, R5 1B 1C
E XRL A, R6 1B 1C
F XRL A, R7 1B 1C
0 JNZ rel 2B 2C
1 ACALL (P3) 2B 2C
2 ORL C, bit 2B 2C
3 JMP @A+DPTR 1B 2C
4 MOV A, #data 2B 1C
5 MOV dir, #data 3B 2C
6 MOV @R0, #data 2B 1C
7 MOV @R1, #data 2B 1C
8 MOV R0, #data 2B 1C
9 MOV R1, #data 2B 1C
A MOV R2, #data 2B 1C
B MOV R3, #data 2B 1C
C MOV R4, #data 2B 1C
D MOV R5, #data 2B 1C
E MOV R6, #data 2B 1C
F MOV R7, #data 2B 1C
0 SJMP rel 2B 2C
1 AJMP (P4) 2B 2C
2 ANL C, bit 2B 2C
3 MOVC A, @A+PC 1B 2C
4 DIV AB 1B 4C
5 MOV dir, dir 3B 2C
6 MOV dir, @R0 2B 2C
7 MOV dir, @R1 2B 2C
8 MOV dir, R0 2B 2C
9 MOV dir, R1 2B 2C
A MOV dir, R2 2B 2C
B MOV dir, R3 2B 2C
C MOV dir, R4 2B 2C
D MOV dir, R5 2B 2C
E MOV dir, R6 2B 2C
F MOV dir, R7 2B 2C
0 MOV DPTR, #data16 3B 2C
1 ACALL (P4) 2B 2C
2 MOV bit, C 2B 2C
3 MOVC A, @A+DPTR 1B 2C
4 SUBB A, #data 2B 1C
5 SUBB A, dir 2B 1C
6 SUBB A, @R0 1B 1C
7 SUBB A, @R1 1B 1C
8 SUBB A, R0 1B 1C
9 SUBB A, R1 1B 1C
A SUBB A, R2 1B 1C
B SUBB A, R3 1B 1C
C SUBB A, R4 1B 1C
D SUBB A, R5 1B 1C
E SUBB A, R6 1B 1C
F SUBB A, R7 1B 1C
0 ORL C, /bit 2B 2C
1 AJMP (P5) 2B 2C
2 MOV C, bit 2B 1C
3 INC DPTR 1B 2C
4 MUL AB 1B 4C
5
6 MOV @R0, dir 2B 2C
7 MOV @R1, dir 2B 2C
8 MOV R0, dir 2B 2C
9 MOV R1, dir 2B 2C
A MOV R2, dir 2B 2C
B MOV R3, dir 2B 2C
C MOV R4, dir 2B 2C
D MOV R5, dir 2B 2C
E MOV R6, dir 2B 2C
F MOV R7, dir 2B 2C
0 ANL C, /bit 2B 2C
1 ACALL (P5) 2B 2C
2 CPL bit 2B 1C
3 CPL C 1B 1C
4 CJNE A, #data, rel 3B 2C
5 CJNE A, dir, rel 3B 2C
6 CJNE @R0, #data,rel 3B 2C
7 CJNE @R1, #data,rel 3B 2C
8 CJNE R0, #data, rel 3B 2C
9 CJNE R1, #data, rel 3B 2C
A CJNE R2, #data, rel 3B 2C
B CJNE R3, #data, rel 3B 2C
C CJNE R4, #data, rel 3B 2C
D CJNE R5, #data, rel 3B 2C
E CJNE R6, #data, rel 3B 2C
F CJNE R7, #data, rel 3B 2C
0 PUSH dir 2B 2C
1 AJMP (P6) 2B 2C
2 CLR bit 2B 1C
3 CLR C 1B 1C
4 SWAP A 1B 1C
5 XCH A, dir 2B 1C
6 XCH A, @R0 1B 1C
7 XCH A, @R1 1B 1C
8 XCH A, R0 1B 1C
9 XCH A, R1 1B 1C
A XCH A, R2 1B 1C
B XCH A, R3 1B 1C
C XCH A, R4 1B 1C
D XCH A, R5 1B 1C
E XCH A, R6 1B 1C
F XCH A, R7 1B 1C
0 POP dir 2B 2C
1 ACALL (P6) 2B 2C
2 SETB bit 2B 1C
3 SETB C 1B 1C
4 DA A 1B 1C
5 DJNZ dir, rel 3B 2C
6 XCHD A, @R0 1B 1C
7 XCHD A, @R1 1B 1C
8 DJNZ R0, rel 2B 2C
9 DJNZ R1, rel 2B 2C
A DJNZ R2, rel 2B 2C
B DJNZ R3, rel 2B 2C
C DJNZ R4, rel 2B 2C
D DJNZ R5, rel 2B 2C
E DJNZ R6, rel 2B 2C
F DJNZ R7, rel 2B 2C
0 MOVX A, @DPTR 1B 2C
1 AJMP (P7) 2B 2C
2 MOVX A, @R0 1B 2C
3 MOVX A, @R1 1B 2C
4 CLR A 1B 1C
5 MOV A, dir 2B 1C
6 MOV A, @R0 1B 1C
7 MOV A, @R1 1B 1C
8 MOV A, R0 1B 1C
9 MOV A, R1 1B 1C
A MOV A, R2 1B 1C
B MOV A, R3 1B 1C
C MOV A, R4 1B 1C
D MOV A, R5 1B 1C
E MOV A, R6 1B 1C
F MOV A, R7 1B 1C
0 MOVX @DPTR, A 1B 2C
1 ACALL (P7) 2B 2C
2 MOVX @R0, A 1B 2C
3 MOVX @R1, A 1B 2C
4 CPL A 1B 1C
5 MOV dir, A 2B 1C
6 MOV @R0, A 1B 1C
7 MOV @R1, A 1B 1C
8 MOV R0, A 1B 1C
9 MOV R1, A 1B 1C
A MOV R2, A 1B 1C
B MOV R3, A 1B 1C
C MOV R4, A 1B 1C
D MOV R5, A 1B 1C
E MOV R6, A 1B 1C
F MOV R7, A 1B 1C